**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

*дисциплина: Администрирование сетевых подсистем*

Студент: Махорин Иван Сергеевич

Студ. билет № 1032211221

Группа: НПИбд-02-21

**МОСКВА**

2023 г.

# Цель работы:

# Целью данной работы является приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

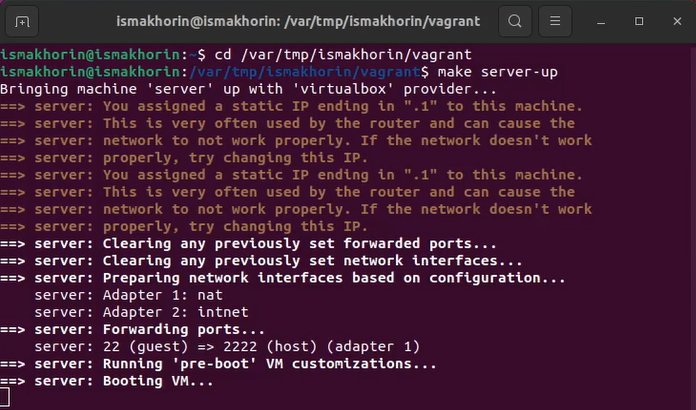
**Выполнение работы:**

Загрузим нашу операционную систему и перейдём в рабочий каталог с проектом:

cd /var/tmp/ismakhorin/vagrant

Далее запустим виртуальную машину server (Рис. 1.1):

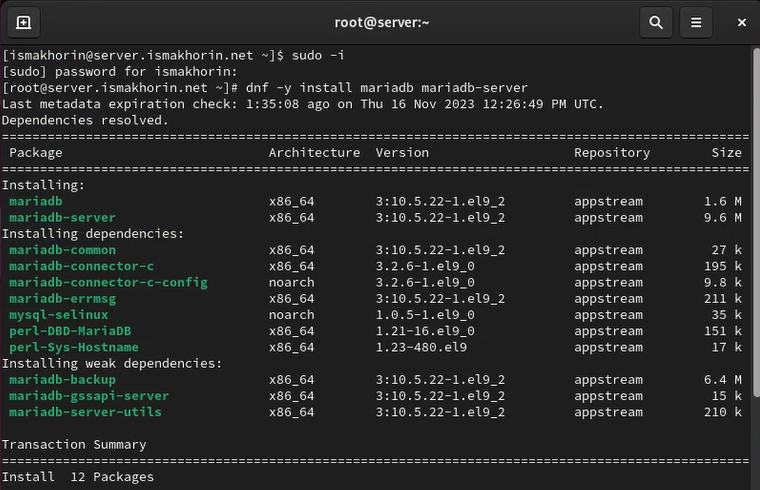
make server-up

****

**Рис. 1.1.** Открытие рабочего каталога с проектом и запуск виртуальной машины server.

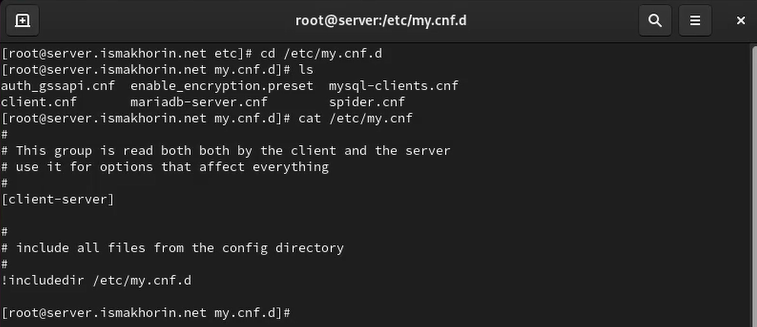
На виртуальной машине server войдём под нашим пользователем и откроем терминал. Далее перейдём в режим суперпользователя и установим необходимые для работы с базами данных пакеты (Рис. 1.2):

dnf -y install mariadb mariadb-server

****

**Рис. 1.2.** Переход в режим суперпользователя и установка необходимых для работы с базами данных пакетов.

Просмотрим конфигурационные файлы mariadb в каталоге /etc/my.cnf.d и в файле /etc/my.cnf (Рис. 1.3):

****

**Рис. 1.3.** Просмотр конфигурационных файлов mariadb в каталоге /etc/my.cnf.d и в файле /etc/my.cnf.

Для запуска и включения программного обеспечения mariadb используем:

systemctl start mariadb

systemctl enable mariadb

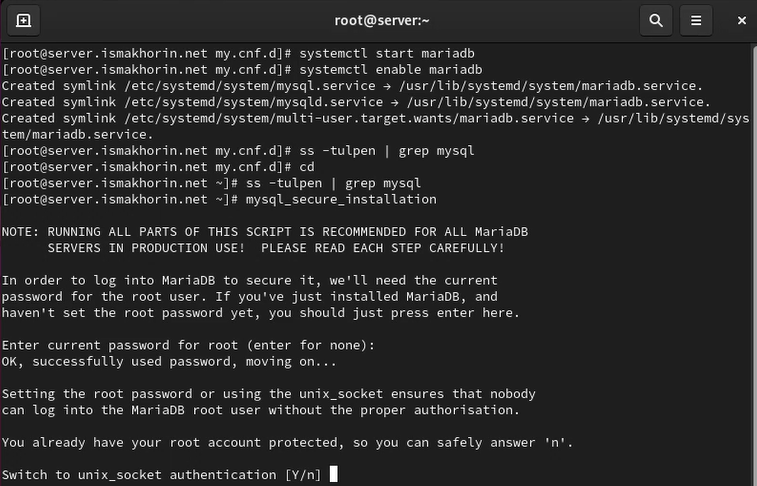
Убедимся, что mariadb прослушивает порт, используя:

ss -tulpen | grep mysql

Теперь мы видим процесс mysqld, прослушивающий порт 3306. После чего запустим скрипт конфигурации безопасности mariadb, используя:

mysql\_secure\_installation

С помощью запустившегося диалога и путём выбора [Y/n] установим пароль для пользователя root базы данных, отключим удалённый корневой доступ и удалим тестовую базу данных и любых анонимных пользователей (Рис. 1.4):

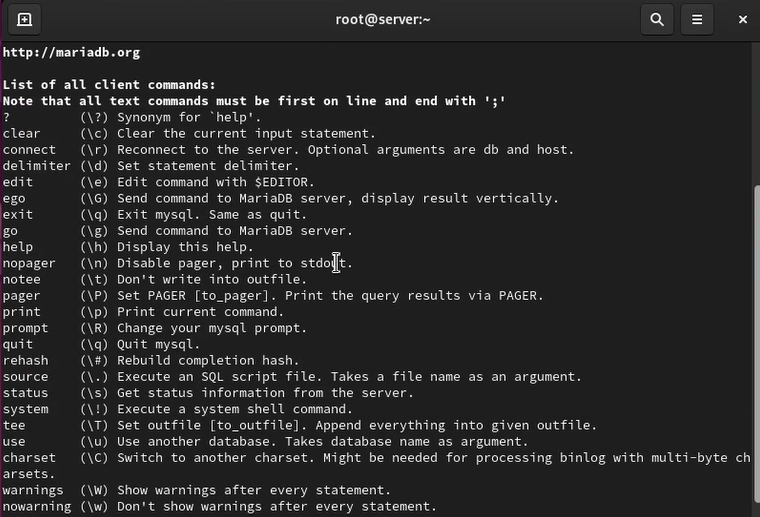
****

**Рис. 1.4.** Запуск и включение программного обеспечения mariadb, проверка прослушивания порта, запуск скрипта конфигурации безопасности mariadb.

Для входа в базу данных с правами администратора базы данных введём:

mysql -u root -p

После чего просмотрим список команд MySQL, введя \h (Рис. 1.5):

****

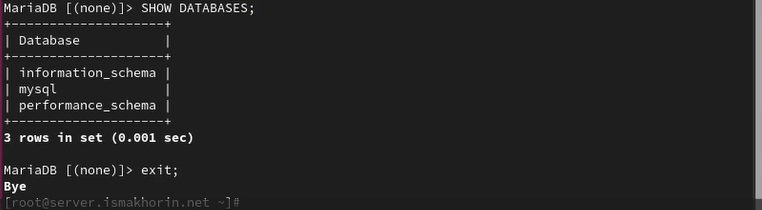
**Рис. 1.5.** Вход в базу данных с правами администратора базы данных и просмотр списка команд MySQL.

Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных введём MySQL-запрос:

SHOW DATABASES;

Для выхода из интерфейса интерактивной оболочки MariaDB введём (Рис. 1.6):

exit;

****

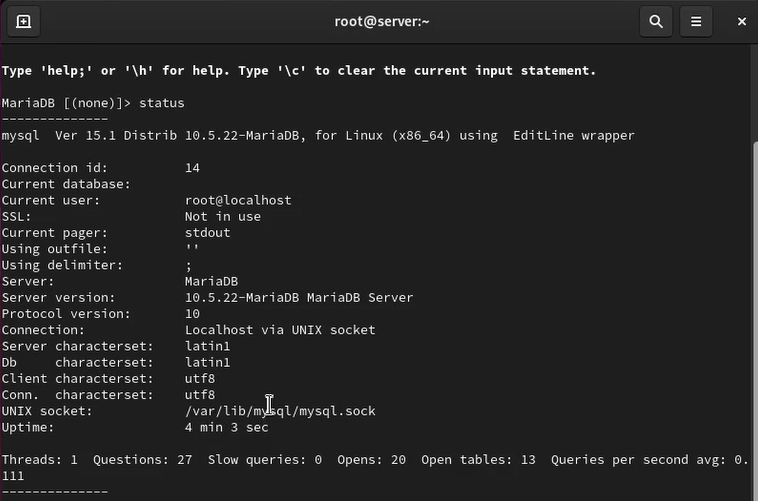
**Рис. 1.6.** Отображение доступных в настоящее время баз данных и выход из интерфейса интерактивной оболочки MariaDB.

Войдём в базу данных с правами администратора:

mysql -u root -p

Для отображения статуса MariaDB введём из приглашения интерактивной оболочки MariaDB (Рис. 2.1):

status

****

**Рис. 2.1.** Вход в базу данных с правами администратора, отображение статуса MariaDB.

В каталоге /etc/my.cnf.d создадим файл utf8.cnf (Рис. 2.2):

cd /etc/my.cnf.d

touch utf8.cnf

****

**Рис. 2.2.** Создание файла utf8.cnf в каталоге /etc/my.cnf.d.

Откроем его на редактирование и укажем в нём следующую конфигурацию (Рис. 2.3):

[client]

default-character-set = utf8

[mysqld]

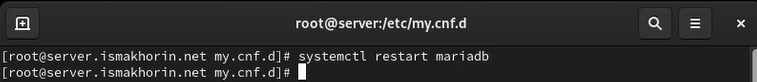
character-set-server = utf8

****

**Рис. 2.3.** Открытие файла на редактирование и указание в нём конфигурации.

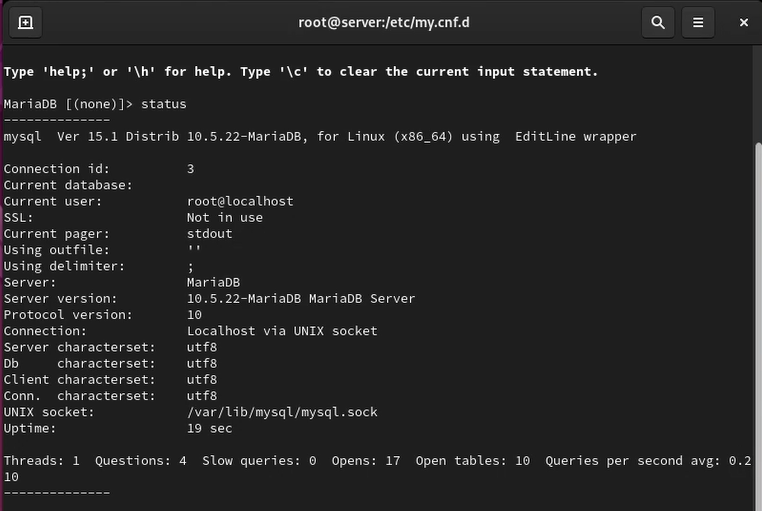
Перезапустим MariaDB (Рис. 2.4):

systemctl restart mariadb

****

**Рис. 2.4.** Перезапуск MariaDB.

Войдём повторно в базу данных с правами администратора и посмотрим статус MariaDB для проверки изменений (рис. 2.5):

****

**Рис. 2.5.** Вход в базу данных с правами администратора и просмотр статуса MariaDB для проверки изменений.

Войдём в базу данных с правами администратора:

mysql -u root -p

Создадим базу данных с именем addressbook:

CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8\_general\_ci;

Теперь перейдём к базе данных addressbook:

USE addressbook;

Отобразим имеющиеся в базе данных addressbook таблицы:

SHOW TABLES;

Создадим таблицу city с полями name и city:

CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));

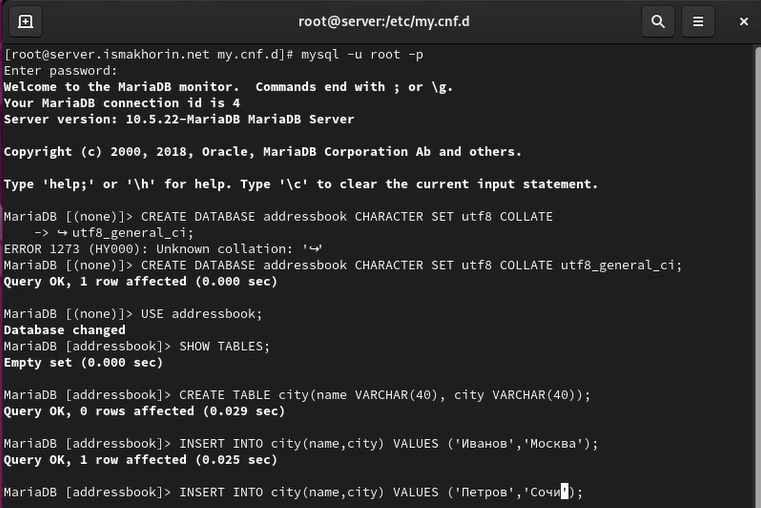
И заполним несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL:

INSERT INTO city(name,city) VALUES ('Иванов','Москва');

В частности, добавим в базу сведения о Петрове и Сидорове (рис. 3.1):

Петров, Сочи

Сидоров, Дубна

****

**Рис. 3.1.** Вход в базу данных с правами администратора, создание базы данных с именем addressbook, открытие базы данных addressbook, отображение имеющиеся в базе данных addressbook таблицы. Создание таблицы city с полями name и city и заполнение таблицы некоторыми данными в соответствии с синтаксисом MySQL.

Сделаем следующий MySQL-запрос:

SELECT \* FROM city;

Теперь создадим пользователя для работы с базой данных addressbook и зададим для него пароль:

CREATE USER ismakhorin@'%' IDENTIFIED BY '**password**';

Предоставим права доступа созданному пользователю ismakhorin на действия с базой данных addressbook (просмотр, добавление, обновление, удаление данных):

GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.\* TO ismakhorin@'%';

Обновим привилегии (права доступа) базы данных addressbook:

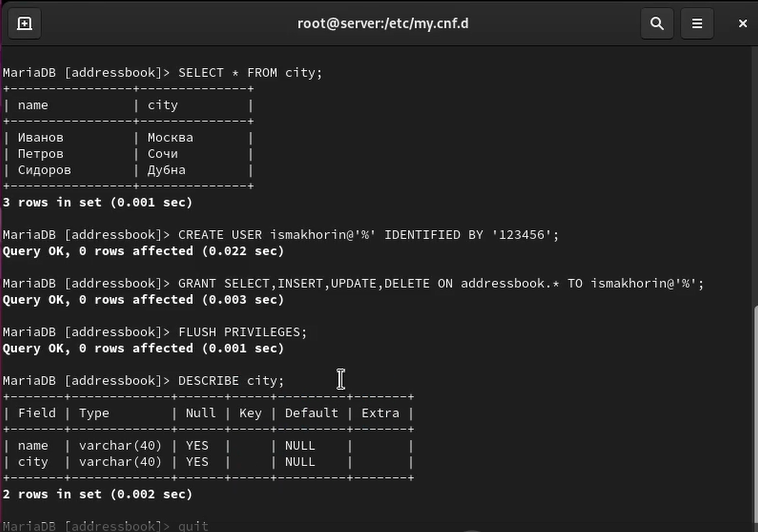
FLUSH PRIVILEGES;

Просмотрим общую информацию о таблице city базы данных addressbook:

DESCRIBE city;

Выйдем из окружения MariaDB (рис. 3.2):

Quit

****

**Рис. 3.2.** MySQL-запрос, создание пользователя для работы с базой данных addressbook, предоставление прав доступа созданному пользователю ismakhorin на действия с базой данных addressbook, обновление привилегии базы данных addressbook, просмотр общей информации о таблице city базы данных addressbook и выход из окружения MariaDB.

Просмотрим список баз данных:

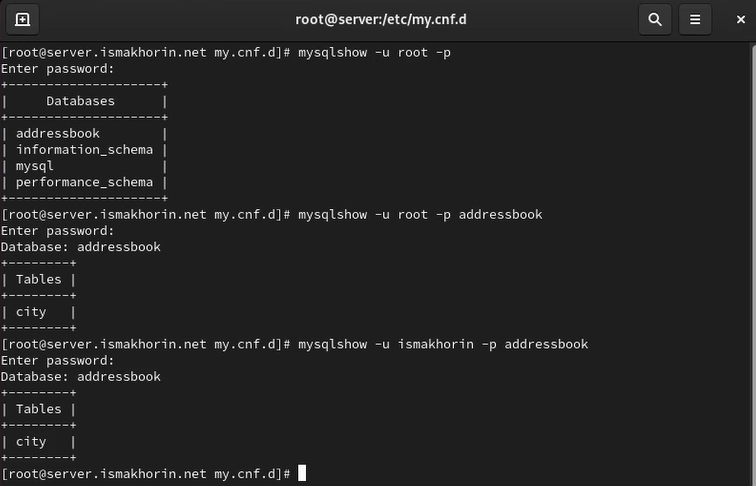
mysqlshow -u root -p

Отдельно просмотрим список таблиц базы данных addressbook (рис. 3.3):

mysqlshow -u root -p addressbook

или

mysqlshow -u ismakhorin -p addressbook

****

**Рис. 3.3.** Просмотр списка баз данных и списка таблиц базы данных addressbook.

На виртуальной машине server создадим каталог для резервных копий:

mkdir -p /var/backup

Сделаем резервную копию базы данных addressbook:

mysqldump -u root -p addressbook > /var/backup/addressbook.sql

Сделаем сжатую резервную копию базы данных addressbook:

mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz

Сделаем сжатую резервную копию базы данных addressbook с указанием даты создания копии:

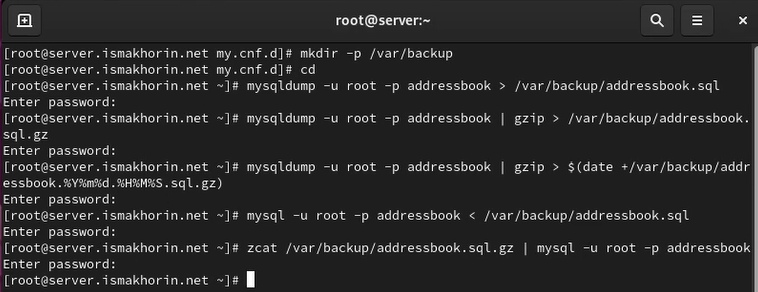
mysqldump -u root -p addressbook | gzip > $(date +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)

Восстановим базу данных addressbook из резервной копии:

mysql -u root -p addressbook < /var/backup/addressbook.sql

Восстановим базу данных addressbook из сжатой резервной копии (рис. 4):

zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook

****

**Рис. 4.** Создание каталога для резервных копий, создание резервной копии базы данных addressbook, создание сжатой резервной копии базы данных addressbook, создание сжатой резервной копии базы данных addressbook с указанием даты создания копии, восстановление базы данных addressbook из резервной копии, восстановление базы данных addressbook из сжатой резервной копии.

На виртуальной машине server перейдём в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог mysql, в который поместим в соответствующие подкаталоги конфигурационные файлы MariaDB и резервную копию базы данных addressbook:

cd /vagrant/provision/server

mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d

mkdir -p /vagrant/provision/server/mysql/var/backup

cp -R /etc/my.cnf.d/utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d/

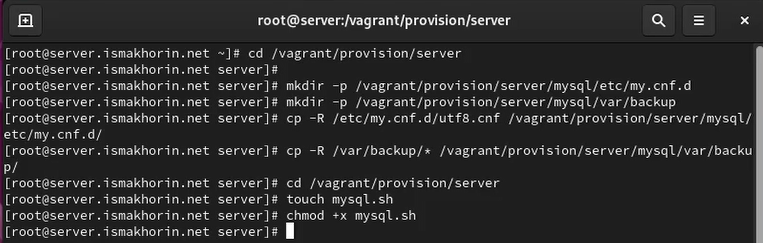
cp -R /var/backup/\* /vagrant/provision/server/mysql/var/backup/

В каталоге /vagrant/provision/server создадим исполняемый файл mysql.sh (рис. 5.1):

cd /vagrant/provision/server

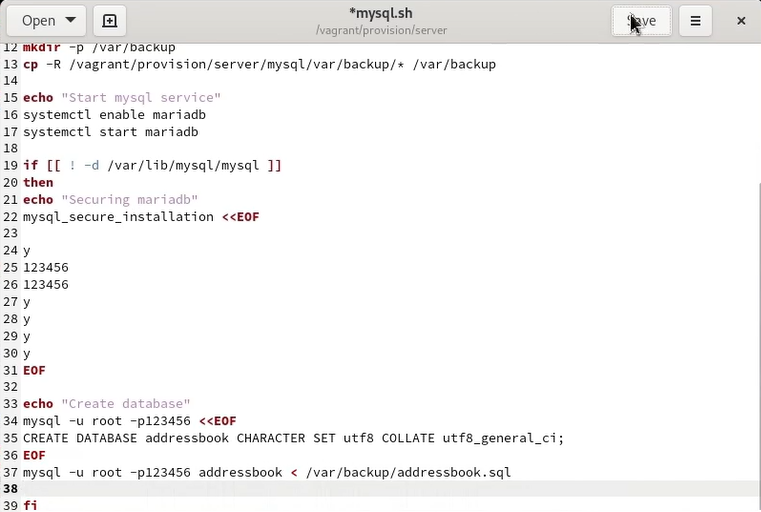
touch mysql.sh

chmod +x mysql.sh

****

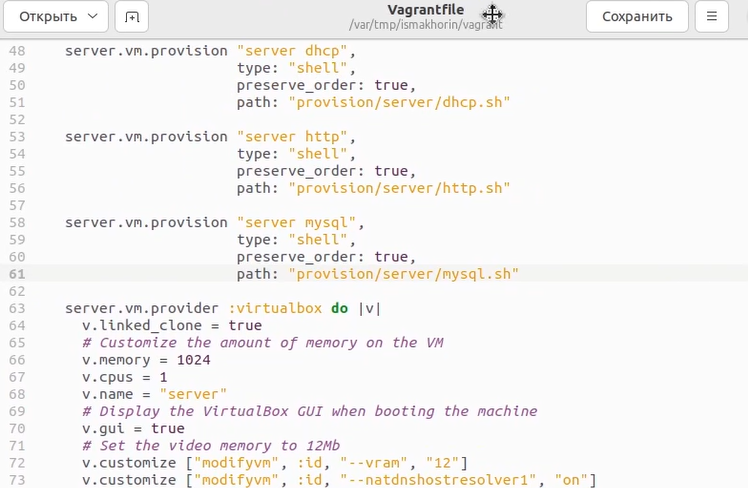
**Рис. 5.1.** Открытие каталога для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога mysql, в который помещаем в соответствующие подкаталоги конфигурационные файлы MariaDB и резервную копию базы данных addressbook. Создание в каталоге /vagrant/provision/server исполняемого файла mysql.sh.

Откроем его на редактирование и пропишем в нём следующий скрипт (рис. 5.2):

****

**Рис. 5.2.** Открытие исполняемого файла на редактирование и прописывание в нём скрипта.

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile добавим в конфигурации сервера следующую запись (рис. 5.3):

****

**Рис. 5.3.** Добавление записи в конфигурационном файле Vagrantfile.

**Вывод:**

# В ходе выполнения лабораторной работы были приобретены практические навыки по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

**Ответы на контрольные вопросы:**

1. Какая команда отвечает за настройки безопасности в MariaDB? - **Настройки безопасности в MariaDB обычно управляются с помощью команды mysql\_secure\_installation. Эта команда выполняет несколько шагов, включая установку пароля для пользователя root, удаление анонимных учетных записей, отключение удаленного входа для пользователя root и удаление тестовых баз данных.**
2. Как настроить MariaDB для доступа через сеть? - **Для настройки MariaDB для доступа через сеть, вы можете отредактировать файл конфигурации MariaDB (обычно называемый my.cnf) и убедиться, что параметр bind-address установлен на IP-адрес, доступный в вашей сети. Также, убедитесь, что пользователь имеет права доступа извне, например, с использованием команды GRANT.**
3. Какая команда позволяет получить обзор доступных баз данных после входа в среду оболочки MariaDB? - **Обзор доступных баз данных после входа в среду оболочки MariaDB можно получить с помощью команды SHOW DATABASES;.**
4. Какая команда позволяет узнать, какие таблицы доступны в базе данных? - **Для просмотра доступных таблиц в базе данных используйте команду SHOW TABLES;.**
5. Какая команда позволяет узнать, какие поля доступны в таблице? - **Чтобы узнать, какие поля доступны в таблице, используйте команду DESCRIBE table\_name; или SHOW COLUMNS FROM table\_name;.**
6. Какая команда позволяет узнать, какие записи доступны в таблице? - **Для просмотра записей в таблице можно использовать команду SELECT \* FROM table\_name;.**
7. Как удалить запись из таблицы? - **Для удаления записи из таблицы используйте команду DELETE FROM table\_name WHERE condition;, где condition - условие, определяющее, какие записи следует удалить.**
8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью? - **Файлы конфигурации MariaDB обычно располагаются в различных местах в зависимости от системы, но основной файл - my.cnf. Он может быть в /etc/my.cnf, /etc/mysql/my.cnf или /usr/etc/my.cnf. С помощью этих файлов можно настроить различные параметры, такие как порт, пути к файлам данных, параметры безопасности и другие.**
9. Где располагаются файлы с базами данных MariaDB? - **Файлы с базами данных MariaDB располагаются в директории данных. Обычно это /var/lib/mysql/ на Linux-системах.**
10. Как сделать резервную копию базы данных и затем её восстановить? - **Для создания резервной копии базы данных используйте команду mysqldump. Например, mysqldump -u username -p dbname > backup.sql. Для восстановления базы данных из резервной копии используйте команду mysql -u username -p dbname < backup.sql.**